Subject: large ArrayCtrl
Posted by hojtsy on Wed, 04 Jan 2006 13:28:52 GMT
View Forum Message <> Reply to Message

Hi,
I have a data table which would fit in a matrix of 20 * 100.000 cells, of approx 50MB size. This seems to be too big to be directly stored in the ArrayCtrl, especially if multiple ArrayCtrls could be opened for the same data table. It would be good to just tell the ArrayCtrl the dimensions of the matrix, and a location of a callback function where the cell data could be accessed on demand. Is this already provided? If not, what would be the simplest way to implement it from client code?
thanks,
Sandor

Subject: Re: large ArrayCtrl
Posted by mirek on Wed, 04 Jan 2006 13:48:56 GMT
View Forum Message <> Reply to Message

Yes. Of course, it is not as trivial as having data directly stored.

The idea is this - instead of storing the "real" data, you use "RowNum" columns. Value of those columns is equal to the index of arrayctrl line.

```
Column&    AddRowNumColumn(const char *text, int w = 0);
```

Now you can provide Display and Convert for those columns. Therefore, the thing you need is to provide Convert that converts your line index to the real data (found elsewhere).

For more columns, you will likely have to one Convert object per column (often you will be able to solve it with single Convert class and attribute, e.g. database column ID).

The last thing to make it work - instead of adding rows to arrayctrl, assing it a "virtual line count".

```
void       SetVirtualCount(int c);
```

Subject: Re: large ArrayCtrl
Posted by hojtsy on Wed, 04 Jan 2006 15:32:44 GMT
View Forum Message <> Reply to Message

Thanks, luzr, this sounds fine. Unfortunately I can not find any documentation for Convert or Display classes. Are these documented somewhere?

Subject: Re: large ArrayCtrl
Posted by mirek on Wed, 04 Jan 2006 16:54:54 GMT
View Forum Message <> Reply to Message

No yet, sorry.

However, Convert perhaps does not even need it - the only thing that is relevant here is

virtual Value Format(const Value& v);

to make your conversion.

Subject: Re: large ArrayCtrl
Posted by hojtsy on Sat, 04 Feb 2006 10:18:14 GMT
View Forum Message <> Reply to Message

I created an example for the big arrays using AddRowNumColumn, SetVirtualCount and Convert. It displays the textual representation of numbers from 1-900000. It may be a candidate for inclusion in the reference dir, since there are no examples for ArrayCtrl with virtual rows.

File Attachments
1) BigArray.zip, downloaded 2348 times

Subject: Re: large ArrayCtrl
Posted by mirek on Sat, 04 Feb 2006 11:10:31 GMT
View Forum Message <> Reply to Message

Good idea. I have changed it a little

```
#include <CtrlLib/CtrlLib.h>

static String sNumberAsText(int number)
{
 static const char * const digits[20] = {
  "", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine",
  "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen",
  "seventeen", "eighteen", "nineteen"
 };
 static const char * const tens[10] = {
  "", "ten", "twenty", "thirty", "fourty", "fifty", "sixty", "seventy", "eighty", "ninety"
 };

 if(number < 20)
  return digits[number];
 if(number < 100)
```

```
  return tens[number / 10] + String(" ") + sNumberAsText(number % 10);
 if(number < 1000)
  return digits[number / 100] + String(" hundred ") + sNumberAsText(number % 100);
 if(number < 1000000)
  return sNumberAsText(number / 1000) + " thousand, " + sNumberAsText(number % 1000);
 return "";
}

struct NumberToText : public Convert {
 virtual Value  Format(const Value& q) const {
  int n = q;
  return n == 0 ? String("zero") : sNumberAsText(n);
 }
};

GUI_APP_MAIN
{
 ArrayCtrl array;
 array.AddRowNumColumn("number", 20);
 array.AddRowNumColumn("text", 80).SetConvert(Single<NumberToText>());
 array.SetVirtualCount(900000);
 TopWindow win;
 win.Zoomable().Sizeable();
 win.Add(array.SizePos());
 win.Run();
}
```

and saved as reference/VirtualArray.

Note that some issues you have done in a little bit complicated way - I am not sure whether you wanted to demostrate other things, however I decided to use to smallest possible code (as I always try for reference examples).

Mirek

Subject: Re: large ArrayCtrl
Posted by hojtsy on Sat, 04 Feb 2006 12:05:02 GMT
View Forum Message <> Reply to Message

This was part of the code that is now written in a shorter form.
array.AddRowNumColumn("number");
array.AddRowNumColumn("text");
array.HeaderTab(0).SetRatio(20);
array.HeaderTab(1).SetRatio(80);
array.ColumnAt(1).SetConvert(numberToText);I see that the current internal logic of the ArrayCtrl makes it necessary to provide two different methods ColumnAt and HeaderTab which return a

different types. But it is hard to remember which method (ColumnAt or HeaderTab) and type (ArrayCtrl::Column or HeaderCtrl::Column) can be used to set/modify/query a specific property of the column. It would be better for the clients to have a single method returning a an object which could be used to manipuate all aspects of the column. I think the naming of these two methods just add to the confusion - both of them are returning a different Column type, but only one is called ColumnAt. Maybe the ArrayCtrl::Column could be extended with all the methods of HeaderCtrl::Column, and those would just forward the call to the HeaderCtrl.

During the development of this example, I found a bug: Assist was not working correctly in the inline constructor of BigArray. It only offered the code completions that are valid in the namespace scope. I believe you can reproduce this if you create a new project with my original example.

Subject: Re: large ArrayCtrl, virtual array count limits
Posted by jaynorwood on Sat, 15 Nov 2008 17:20:32 GMT
View Forum Message <> Reply to Message

I'm experiencing limits in the VirtualArray example beyond which there is no refresh. It works for 100million, but not 200million virutal array count in its current form.

I attempted changing some parameters to int64, but the limit still occurs. Below is my modified code.

I'm a bit surprised at the 200 million limit. I thought perhaps a 2Gig limit in the original due to the signed int parameter range.

Aside from figuring out what this strange 200 million limit is about, I'd like to see the framework improved so that int64 ranges could be used throughout.

```
#include <CtrlLib/CtrlLib.h>

using namespace Upp;

static String sNumberAsText(int64 number)
{
 static const char * const digits[20] = {
  "", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine",
  "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen",
  "seventeen", "eighteen", "nineteen"
 };
 static const char * const tens[10] = {
  "", "ten", "twenty", "thirty", "fourty", "fifty", "sixty", "seventy", "eighty", "ninety"
 };

 if(number < 20)
  return digits[number];
 if(number < 100)
  return tens[number / 10] + String(" ") + sNumberAsText(number % 10);
```

```
 if(number < 1000)
  return digits[number / 100] + String(" hundred ") + sNumberAsText(number % 100);
 if(number < 1000000)
  return sNumberAsText(number / 1000) + " thousand, " + sNumberAsText(number % 1000);
 if(number < 1000000000)
  return sNumberAsText(number / 1000000) + " million, " + sNumberAsText(number % 1000000);
 if(number < 1000000000000ULL)
  return sNumberAsText(number / 1000000000) + " billion, " + sNumberAsText(number %
1000000000);
 return "";
}

struct NumberToText : public Convert {
 virtual Value  Format(const Value& q) const {
  int64 n = q;
  return n == 0 ? String("zero") : sNumberAsText(n);
 }
};

GUI_APP_MAIN
{
 ArrayCtrl array;
 array.AddRowNumColumn("number", 20);
 array.AddRowNumColumn("text", 100).SetConvert(Single<NumberToText>());
 array.SetVirtualCount(200000000);
 TopWindow win;
 win.Zoomable().Sizeable();
 win.Add(array.SizePos());
 win.Run();
}
```

Subject: Re: large ArrayCtrl, virtual array count limits
Posted by mirek on Sun, 16 Nov 2008 17:58:37 GMT
View Forum Message <> Reply to Message

jaynorwood wrote on Sat, 15 November 2008 12:20I'm experiencing limits in the VirtualArray example beyond which there is no refresh. It works for 100million, but not 200million virutal array count in its current form.

Well, not that is I would call "pushing to the limits"

Quote:
I'm a bit surprised at the 200 million limit. I thought perhaps a 2Gig limit in the original due to the signed int parameter range.

IMO, this is caused because of the height of line in pixels. That is about 15 pixels, means maximum is close to 2G / 15.

Quote:
Aside from figuring out what this strange 200 million limit is about, I'd like to see the framework improved so that int64 ranges could be used throughout.

Uh, that would work only as long as somebody does not decide to test int64 limits IMO.

Seriously, if your real world application needs 200 millions of lines in table TO BE PRESENTED AT ONCE, something is wrong with your UI design.

Also, realistically, any nontrivial usage would require those data to come from somewhere, and that would take eons.

Making these limits int64 would meant making EVERYTHING 64bit, with ugly impacts on code compactness and speed.

Mirek

Subject: Re: large ArrayCtrl, virtual array count limits
Posted by jaynorwood on Sun, 16 Nov 2008 19:47:23 GMT
View Forum Message <> Reply to Message

Well, we have a large database ... perhaps 4G records ... being generated from microprocessor trace data, and I thought perhaps your virtual array control could be used as a front end to move around in the data in a flat mode. This limitation on the scrolling range seems rather arbitrary with as much template parameter support as there is in other areas of the code. I don't see why the scrolling limits couldn't be just another template parameter.

I seem to recall that eclipse jface or swt even supports bignum scrolling ranges with their virtual array, so it isn't like this is an out of the ordinary expectation for the virtual array.
